REMARKS

Applicants wish to thank the Examiner for his telephonic interview January xxx, 2008. Per our discussion, we believe the claims as presented are in condition for allowance.

In the instant Office Action, the Examiner rejected claims 55 and 56 under 35 USC 103(a) as unpatentable over U.S. Patent 6,343,339 to Daynes (hereinafter "Daynes") in view of the excerpt from the book, "Structured Computer Organization" by Andrew S. Tanenbaum (hereinafter "Tanenbaum"). Claims 1-17 and 57 were rejected under 35 U.S.C. 103(a) as being unpatentable over Daynes in view of Tanenbaum, and U.S. Patent 6,535,968 to Pham (hereinafter "Pham").

Tanenbaum is a textbook typically used to teach college or graduate students various computing principles as they relate to hardware, software, operating systems and their use in computing. In particular, Tanenbaum states that "Hardware and software are logically equivalent" as a concept to keep in mind while reading the remainder of the book. It is the opinion of Tanenbaum that software can be implemented in hardware if desired and that hardware can be simulated in software. Applicants do not disagree with the concept that there is a relationship between hardware and software.

However, Tanenbaum does not teach or suggest any particular hardware or software design. It also does not teach or suggest that a software design can be exactly implemented in hardware or that a hardware design can exactly be implemented in software. In fact, Tanenbaum

admits that hardware can be **simulated** in software which often means that there is not an exact translation between the hardware and the software due to the overhead and inexactness of the simulation. For example, a shift register could be simulated in software using hardware simulation software like SPICE or other simulation environments. Unfortunately, the simulation requires a hardware design to be simulated but alone cannot teach or suggest anything in particular. Further, these simulation environments do not operate with the same performance nor do they achieve the same design goals for speed, efficiency and/or accuracy. Also, many of the timing issues in hardware cannot be simulated due to fabrication and other issues that only become known when initial chip designs have been fabricated and tested. For at lease these reasons, we believe Tanenbaum does not apply to the claims as originally filed or in light of our clarification of the current claims as presented herein.

Daynes describes creating a locking protocol for transactions that need certain computing resources. (Col. 3, lines 65-66 to Col. 4, lines 1-4) Each resource in Daynes has a lock with various associated lock states. (*Id.*) The table of immutable lock states (TILS) records all of the lock states considered "immutable" or unchangeable as a result of a lock operation in the locking protocol. (Col. 4, lines 12-18) According to Daynes, certain operations may acquire a resource without modifying the associated lock state while other lock operations may require the lock state for a resource to be modified or changed due to a transaction's use of the resource. (Col. 4, lines 5-11) The TILS is a table in Daynes loaded with the lock states for a particular resource and used to avoid duplication of the lock states for a given resource. (*Id.*) For example, the TILS in

Daynes describes a table that allows a given resource to be shared with multiple transactions as long as the lock state for the resource required by each transaction is the same. (*Id.*)

Each transaction in Daynes has a lock set describing the set of locks and resources used by the transaction. When the transaction ends, the set of locks in the lock set are released thus allowing the resources to be used by other transactions. (Col. 4, lines 51-55). Daynes indicates that "a stack" can be used to store these locks for each transaction. A more simple approach suggested in Dayness requires that the TILS is scanned directly to remove locks and free resources associated with the terminated transaction.

Unfortunately, Daynes alone or in combination with Tanenbaum and Pham does not teach or suggest in software and/or hardware, "a hash bucket memory having an array of bit-bucket pointers that each operate as a head pointer to a linked list of active semaphore structures, the hash bucket memory addressable by a hash address derived through application of a hash function to a semaphore value and wherein each semaphore structure accommodates a current thread that owns a semaphore and potentially one or more waiting threads in a queue waiting for release of the semaphore" as recited in claim 1 as amended. To establish a prima facie obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. In re Royka , 490 F.2d 981, 180 USPQ 580 (CCPA 1974). To make a prima facie case of obviousness, the Examiner must determine the "scope and content of the prior art," ascertain the "differences between the prior art and the claims at issue," determine "the level of ordinary skill in the pertinent art," and evaluate evidence of secondary considerations. Graham v. John

Deere, 383 U.S. 1, 17, (1966); KSR Int'l Co. v. Teleflex Inc., 550 U.S. ___ (2007); see also M.P.E.P. § 2141.

Daynes briefly mentions that certain lock requests are added to a queue but does not provide any details on implementing the queue in either hardware or software. Indeed, Daynes does not describe or suggest a particular organization of semaphores in a linked list or a method for addressing the semaphores in such a linked list with a hashing. Claim 1 recites using a specific "hash bucket memory" in a hardware implementation as it is both flexible to accommodate many semaphores and locks but is fast as it is implemented in hardware. Daynes mentions using a hash function for a simple table but the table is not equivalent to the linked list and organization of current threads with waiting threads.

Daynes alone or in combination with Tanenbaum and Pham also does not teach or suggest, "semaphore circuitry, coupled to said ZBT interface, that receives a signal from said network processor, and that controls said semaphore associated with the corresponding predetermined linked list of active semaphore structures in the hash bucket memory wherein the received signal and said semaphore are related to locking and unlocking access to data in the storage environment" as also recited in claim 1 as amended. Indeed, Daynes mentions the existence of the Internet yet does not suggest separating the functionality of processing semaphores with semaphore circuitry on a separate co-processor that communicates with a network processor. Instead, the Internet in Daynes is mentioned as a medium that "Computer 100" can send messages and data to other Internet users. (Col. 6, lines 50-55 of Daynes). Just

mentioning the Internet in Daynes does not teach or suggest separating a network processor from semaphore circuitry and processing semaphores over a bus or network.

Likewise, Pham describes a ZBT for data transfers in general but does not teach or suggest using the ZBT in conjunction with semaphores. The Supreme Court in KSR stated that it is "important [for an examiner] to identify a reason that would have prompted a person of ordinary skill in the relevant field to combine the [prior art] elements" in the manner claimed. *KSR Int'l Co. v. Teleflex, Inc.*, No. 04-1350, slip op. at 14 (U.S. April 30, 2007). The Court indicated that there should be an *"explicit"* analysis regarding "whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue." Id. (emphasis added). Further, the Court did not reject the use of "teaching, suggestion, or motivation" test as a factor in the obviousness analysis, but rather stated that this test may be indicative of non-obviousness under 35 U.S.C. § 103. Id. at 14-15. In the instant office action there is no explicit analysis providing a suggestion to combine or teaching the use of a ZBT in Pham to separate a network processor from semaphore circuitry or logic.

Daynes also does not teach or suggest alone or in combination with Tanenbaum and Pham, "an update engine, further included in the semaphore circuitry, that upon receipt of a received signal from said network processor relating to a thread on said network processor, processes said semaphore related to said received signal and responds, as required, by sending a transmitted signal back to said network processor in association with said semaphore" as recited in claim 1 as amended. As previously described, Daynes does not separate a network processor

from semaphore circuitry or processing over a bus or network. Daynes is silent on the issue of processing semaphores with a co-processor let alone transmission of locks or semaphores between an update engine included in the semaphore circuitry and a network processor. For at least this additional reason, claim 1 is in condition for allowance.

Dependant claims 2-17 are independently allowable as well as allowable by virtue of their dependence on allowable independent claim 1 for at least the reason described previously. Accordingly, Applicant respectfully submits dependant claims 2-17 are also in condition for allowance.

Further, Daynes alone or in combination with Tanenbaum does not teach or suggest claim 55 as currently amended for at least some if not all of the reasons described in conjunction with claim 1. Claims 56-57 are not only allowable independently but also allowable by virtue of their dependence on claim 55.

For at least these reasons, independent claims 1 and 55 as currently filed are in condition for allowance. Dependent claims 2-17 and 55-57 are allowable independently as well as by virtue of their direct or indirect dependency on claims 1 and 55 respectively.

Applicants have made a diligent effort to place the aforementioned claims in condition for allowance. Accordingly, Applicants respectively request a withdrawal of the rejections and immediate allowance of claims 1-17 and 55-57. Of course, should there remain unresolved issues or the Examiner believes a discussion appropriate, it is respectfully requested that the

Applicant(s): Herbst, et al. Application No.: 10/696,624 Filed: 10/23/2003 Title: Apparatus to Offlload and Accelerate Pico Code Processing Running in a Storage Processor Attorney Docket No.: 00121-002900000	Group Art Unit: 2187 Examiner: Brian R. Peugh

Examiner telephone Leland Wiesner, Applicants' Attorney at (650) 853-1113 so that such issues may be resolved as expeditiously as possible.

Respectfully Submitted,

02/01/2008
Date

Leland Wiesner
Attorney for Applicants
Reg. No. 39424

Leland Wiesner
Wiesner and Associates
366 Cambridge Avenue
Palo Alto, California 94306
Tel. (650) 853-1113